UNIVERSITÀ DEGLI STUDI DI MILANO

Architettura degli Elaboratori e delle Reti
Turno 2 – Prof. Federico PEDERSINI
I prova in itinere – 30 aprile 2009

Cognome, nome: Matricola:

Per superare la prova è necessario totalizzare almeno 18 punti, di cui almeno 9 negli esercizi 6 e 7.

1. [1.5] Si converta il numero **0xFC18**: a) in base 2 e b) in numero decimale supponendo che si stia utilizzando la notazione in complemento a 2 per numeri negativi.

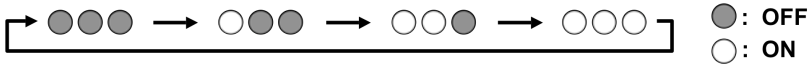
2. [1.5] Si rappresenti il numero **–13,6875** in formato IEEE-754 – singola precisione.

3. [3] Si disegni la struttura interna di un full-adder. Si disegni la struttura di un sommatore a propagazione di riporto a 4 bit e se ne calcoli il cammino critico.

4. [4] Si progetti un circuito caratterizzato da un ingresso a 4 bit rappresentante un numero binario intero senza segno **A**, e un'uscita che vale '1' se e solo se:
(A<4 ed è divisibile per 2) oppure (4≤A<8) oppure (A≥8 ed è divisibile per 4).

a) Determinare la tabella di verità della funzione logica di uscita;
b) scrivere la funzione nella forma canonica più adatta;
c) semplificarla mediante mappa di Karnaugh;

5. [9] Si sintetizzi una macchina a stati finiti (di Moore) che comanda un display costituito da 3 lampade, le quali vengono accese in modo da produrre, in sequenza, le configurazioni mostrate in figura (si consideri la corrispondenza: uscita "1" = lampada "ON", uscita "0" = lampada "OFF").



La macchina passa da una configurazione alla successiva ogni qualvolta si presenta un fronte di salita sull'ingresso **I** della macchina. Il valore dell'ingresso viene controllato ogni centesimo di secondo.

Si determinino STG, STT, STT codificata e struttura circuitale del sistema completo, avendo cura di semplificare il più possibile le funzioni prima di tradurle in circuito.

6. [10] Si traducano in linguaggio Assembly le seguenti procedure. Si seguano le convenzioni MIPS per l'utilizzo dei registri argomento e valore delle funzioni.

```
int funzUno( unsigned int x )
{
   if( x < 5 )
      return( funzDue(x) );
   else
      return( x + funzUno(x-5) );
}
```

```
int funzDue( unsigned int x )
{
      return( x*x );
}
```

7. [6] Si traducano le seguenti pseudoistruzioni: **a)** in Assembly MIPS nativo e **b)** in linguaggio macchina MIPS, specificando anche la dimensione in bit dei campi dell'istruzione.

```
li $t0, 65600
divi $t0, $t1, 5
lw $s2, $s1($s0)
bgei $a0, -20, -20   # (branch on greater or equal than immediate)
```


Registri MIPS *general purpose*:

0	zero		24-25	t8 – t9
1	at		26-27	k0 – k1
2-3	v0 – v1		28	gp
4-7	a0 – a3		29	sp
8-15	t0 – t7		30	s8
16-23	s0 – s7		31	ra

MIPS Instruction Set: